Building Asynchronous SNMP Agents

Presented by Ilya Etingof <etingof@gmail.com>

Why SNMP

- SNMP is old, complicated and has many competitors
- SNMP is still ubiquitous in monitoring
- SNMP is well-understood by many
- We have accumulated over 10,000 MIBs [1]

1. http://mibs.snmplabs.com/asn1/

Consider the use-case

- Your network is SNMP-monitored
- New hardware arrives
- Being new, it offers just REST API...

How'd your NMS reach it?

Standing up a mediation proxy

1. Pick a MIB (e.g. *SNMPv2-MIB*) or come up with your own
2. Turn the *SNMPv2-MIB* into Python code with hooks
3. Interface MIB hooks with your RESTful server
4. Fire up the SNMP agent

Demo: Redfish as a data source

```
$ curl http://demo.snmplabs.com/redfish/v1/Systems/437XR1138R2
...
{
 "Id": "437XR1138R2",
 "Name": "WebFrontEnd483",
 "SystemType": "Physical",
 "AssetTag": "Chicago-45Z-2381",
 "Manufacturer": "Contoso",
 "Model": "3500",
 "SubModel": "RX",
 "SKU": "8675309",
 "SerialNumber": "437XR1138R2",
 "PartNumber": "224071-J23",
 "Description": "Web Front End node",
 "UUID": "38947555-7742-3448-3784-823347823834",
 "HostName": "web483",
 ...
```

Demo: system name

```
$ ls -l SNMPv2-MIB.txt
-rw-r--r--   1 ietingof  staff     29305 Jul 24 07:54 SNMPv2-MIB.txt
$ cat SNMPv2-MIB.txt
...
sysName OBJECT-TYPE
     SYNTAX      DisplayString (SIZE (0..255))
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
             "An administratively-assigned name for this managed
             node.  By convention, this is the node's fully-qualified
             domain name.  If the name is unknown, the value is
             the zero-length string."
     ::= { system 5 }
...
```

Demo: Pythonize SNMPv2-MIB

```
$ mibdump --destination-format pysnmp --destination-template \
    pysnmp/mib-instrumentation/managed-objects-instances.j2 SNMPv2-MIB
$ ls -l SNMPv2-MIB.py
-rw-r--r--   1 ietingof  staff     16839 Jul 24 07:54 SNMPv2-MIB.py
```

Looking inside the *SNMPv2-MIB.py*:

```
...
class SysnameObjectInstance(MibScalarInstance):
    def readTest(self, varBind, **context):
        # Put your code here
        MibScalarInstance.readTest(self, varBind, **context)

    def readGet(self, varBind, **context):
        # Put your code here
        MibScalarInstance.readGet(self, varBind, **context)
...
```

Demo: add REST API call

```
REST_API_URL = 'http://demo.snmplabs.com/redfish/v1/Systems/437XR1138R2'

executor = concurrent.futures.ThreadPoolExecutor()

def readGet(self, (name, value), **context):
    cbFun = context['cbFun']

    def done_callback(future):
        rsp = future.result()
        value = self.syntax.clone(rsp['HostName'])
        cbFun((name, value), **context)

    future = executor.submit(load_url, REST_API_URL)

    future.add_done_callback(done_callback)
```

Demo: stand up SNMP agent

Configure SNMP Command Responder:

```
$ pip install snmpresponder
$ cp SNMPv2-MIB::sysName.py /etc/snmpresponder/managed-objects/
$ snmpresponderd
```

And query it:

```
$ snmpget -v2c -c public localhost SNMPv2-MIB::sysName.0
SNMPv2-MIB::sysName.0 = STRING: web483
```

Why it all matters

- SNMP is still widely used in monitoring
- But data sources may vary
- The *snmpresponder* [1] tool offers universal mediation layer

1. https://github.com/etingof/snmpresponder

Thank you

;-)